CLAIMS

What is claimed is:

1. A system comprising:

a first node including data having an associated dirty (D) state; and

a second node operative to provide a source broadcast requesting the data, the first node being operative in response to the source broadcast to provide the data to the second node and to transition the state associated with the data at the first node from the D-state to an owner (O) state without concurrently updating memory, a shared (S) state being associated with the data at the second node.

2. The system of claim 1, wherein the first node is operative to transition the state associated with the data at the first node from the D-state to a modified (M) state in connection with writing the data to the first node prior to receiving the source broadcast from the second node.

3. The system of claim 2, wherein the first node is operative in response to the source broadcast to provide the data to the second node and transition the state associated with the data at the first node from the M-state to an invalid state without updating the memory, a D-state being associated with the data at the second node in response to receiving the data from the first node.

4. The system of claim 3, wherein the data at the second node having an associated D-state is available for migration to other nodes.

5. The system of claim 1, wherein the system further comprises another node that includes the data having an associated modified (M) state associated with the data prior to the first node including the data, the data migrating from the another node to the first node in response to the first node providing a source broadcast read request for the data, the data having the associated D-state in the first node in response to the data migrating from the another node to the first node.

6. The system of claim 5, wherein the another node having the data in the associated M-state transitions from the M-state to an invalid state without updating memory after providing the data to the first node.

7. The system of claim 1, wherein further migration of the data from the second node is precluded.

8. The system of claim 1, further comprising at least one other node that provides a non-data response in response to the source broadcast request from the second node indicating that the at least one other node does not have a valid copy of the data requested by the second node.

9. The system of claim 1, wherein the first node defines a first processor and the second node defines a second processor, the first and second processors each having an associated cache that comprises a plurality of cache lines, each cache line having a respective tag address that identifies associated data and each cache line having state information that indicates a state of the associated data for the respective cache line, the first and second processors being capable of communicating with each other and with a system memory via an interconnect.

10. The system of claim 9, further comprising a first cache controller associated with the first processor and a second cache controller associated with the second processor, the first cache controller being operative to manage data requests and responses for the associated cache of the first processor, the first cache controller effecting state transitions associated with the data in the associated cache of the first processor based on the data requests and responses for the associated cache of the first processor, the second cache controller being operative to manage data requests and responses for the associated cache of the second processor, the second cache controller effecting state transitions associated with the data in the associated cache of the second processor based on the data requests and responses for the associated cache of the second processor.

11. The system of claim 9, wherein the system implements a hybrid cache coherency protocol wherein each of the first and second processors employs a source broadcast-based protocol to issue a request for the data and provide responses for the data, and employs an associated forward progress protocol to reissue a request for the data in response to the request failing in the source broadcast protocol.

12. A multiprocessor network comprising:

memory for storing data;

a plurality of processor nodes in communication with each other and with the memory;

a first processor node of the plurality of processor nodes including data in a cache line having an associated dirty state; and

a second processor node of the plurality of processor nodes operative to provide a source broadcast read request to obtain the data;

the first processor node being programmed to respond to the source broadcast read request of the second processor node by providing a shared data response to the second processor node and transitioning the data in the cache line from the dirty state to an owner state without concurrently updating the memory with the data, the data being stored in a cache line at the second processor node associated with a shared state.

13. The network of claim 12, wherein the first processor node is further programmed to transition the state associated with the data in the cache line of the first processor node from the dirty state to a modified state in connection with writing the data at the first processor node.

14. The network of claim 13, wherein the first processor node is operative while in the modified state to respond to a source broadcast read request for the data from a requester node by providing an ownership data response to the requester and to invalidate the modified data at the first processor node without updating the memory, the data being associated with a given address at the requester and having an associated dirty state in response to receiving the ownership data response from the first processor node.

15. The network of claim 12, wherein the first processor node is operative to receive the data and have an associated dirty state in response to the first processor providing a source broadcast requesting the data, the dirty data at the first processor node being provided by a processor node including the data and having a modified state associated with the data.

16. The network of claim 14, further comprising a third processor node that provides the data to the first processor node prior to the first processor node including the data in

the cache line in the dirty state, the third processor node including the data in a modified state and providing an ownership data response to the first processor node such that the first processor node includes the data in the cache line in the associated dirty state, the third processor node transitioning the data in the cache line from the dirty state to an invalid state without updating the memory after providing the ownership data response to the first processor node.

17. The network of claim 12, wherein the network implements hybrid cache coherency protocol that employs a source broadcast protocol to process manage source broadcast requests provided by nodes within the network and, if a request fails, the network transfers to an associated forward progress protocol.

18. The network of claim 17, wherein the forward progress protocol comprises a directory-based protocol.

19. A computer system comprising:

a source processor having an associated cache, the source processor operative to issue a source broadcast request for data;

memory storing the data; and

a target processor having an associated cache with a cache line that includes the data, the cache line having an associated dirty (D) state, the target processor being programmed to (i) when not writing the data in the cache line of the target processor, respond to the source broadcast request by providing a shared data response to the source processor and by transitioning the state of the data in the cache line from the D-state to an owner (O) state without concurrently updating the memory, and (ii) programmed to transition the state of the data in the cache line from the D-state to a modified (M) state in connection with writing the data in the cache line of the target processor, and the first processor node being operative while in the M-state to respond to the source broadcast request by providing an ownership data response to the another processor node and by transitioning the state of the data in the cache line from the M-state to an invalid (I) state without updating the memory.

20. The computer system of claim 19, wherein the source processor further comprises a cache having a cache line for storing the data, the source processor storing the data in the cache line of the source processor in a shared state in response to receiving the shared data response from the target processor.

21. The computer system of claim 19, wherein the source processor further comprises a cache having a cache line for storing the data, the source processor storing the data in the cache line of the source processor in the D-state in response to receiving the ownership data response from the target processor.

22. The computer system of claim 19, further comprising a third processor having an associated cache and a cache line that includes the data and has an associated M-state prior to the target processor storing the data in the cache line in the D-state, the third processor being operative to provide an ownership data response to the target processor in response to a source broadcast read request from the target processor for the data and to transition the data in the cache line of the third processor from the M-state to the I-state, the data being placed in the cache line of the target processor cache in the D-state.

23. The computer system of claim 19, wherein the computer system implements hybrid cache coherency protocol that employs a source broadcast protocol that defines rules for processing broadcast requests provided by processors within the system, if a request fails using the source broadcast protocol, the system transfers to an associated forward progress directory-based protocol.

24. A system comprising:

means for broadcasting from a first node a request for data;

means for providing the data from a second node to the first node, the data at the second node having an associated dirty (D) state, a shared (S) state being associated with the data at the first node in response to the first node receiving the data from the second node; and

means for transitioning the state associated with the data at the second node from the D-state to an owner (O) state without concurrently updating memory of the system.

25. The system of claim 24, further comprising:

means for transitioning the data in the associated cache line at the second node from the D-state to a modified (M) state to permit the second node to write the data in an associated cache line at the second node;

means for providing a response from the second node to the first node that includes the data and for invalidating the data at the second node without updating memory; and

means for associating a D-state with the data at the first node in response to receiving the response from the second node.

26. The system of claim 24, further comprising means for providing the data from another node to the second node, the D-state being associated with the data at the second node in response to the second node receiving the data from the another node.

27. The system of claim 26, wherein the another node includes the data in modified (M) state, the system further comprising means for transitioning the state associated with the data at the another node from the M-state to an invalid state without writing-back to memory after the another node provides the data to the second node.

28. A method comprising:

broadcasting a request for data from a first node to other nodes of an associated system;

transitioning a state associated with the data at a second node from a dirty (D) state to an owner (O) state data without concurrently updating memory if the second node does not require to modify the data;

responding to the request broadcast from the first node by providing a response from the second node that includes a shared copy of the data; and

transitioning the state associated with the data at the first node to a shared (S) state in response to receiving the response from the second node.

29. The method of claim 28, further comprising:

obtaining permission for the second node to write the data to a cache line at the second node;

transitioning the state associated with the data at the cache line of second node to a modified (M) state;

providing the response from the second node to the first node;

transitioning the state associated with the data at a cache line of the first node to the D-state; and

invalidating the data at the cache line of second node without updating memory.

30. The method of claim 28, wherein the associated system defines a multiprocessor system and the first and second nodes comprises processors in the multiprocessor system, the method further comprising:

employing a source broadcast protocol that defines rules for processing broadcast requests provided by the processors and the memory within the multiprocessor system; and

reissuing a given request using an associated forward progress protocol if the given request fails while employing the source broadcast protocol.

31. A computer system comprising a cache coherency protocol that is operative to permit read migration of data to a cache associated with a source processor from a cache associated with a target processor when the data is written to the cache associated with the target processor, the protocol being further operative to prevent future read migration of the data to the cache associated with the source processor from the cache associated with the target processor when the data is not written to the cache associated with the target processor.